Floating Point Exponent (ALTFP_EXP)

Megafunction User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Document Version: 1.0
Document Date: December 2008

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

UG-01053-1.0

I.S. EN ISO 9001

Contents

Chapter 1. About this Megafunction

Introduction . . . 1–1
Features . . . 1–1
General Description . . . 1–1
Floating-Point Format . . . 1–2
Special Case Numbers . . . 1–3
Rounding . . . 1–3
Exception Handling . . . 1–3
Truth Table . . . 1–4
Resource Utilization and Performance . . . 1–4

Chapter 2. Getting Started

MegaWizard Plug-In Manager Page Descriptions . . . 2–1
Design Example: Exponential of Single Precision Numbers . . . 2–4
Design Files . . . 2–4
Example . . . 2–4
Functional Simulation in the ModelSim-Altera Software . . . 2–5
Understanding the Simulation Results . . . 2–6

Chapter 3. Specifications

Ports and Parameters . . . 3–1
Input Ports . . . 3–1
Output Ports . . . 3–1
Parameters . . . 3–2

Additional Information

Document Revision History . . . Info–1
Referenced Documents . . . Info–1
How to Contact Altera . . . Info–1
Typographic Conventions . . . Info–2

1. About this Megafunction

Introduction

As design complexities increase, the use of vendor-specific intellectual property (IP) blocks has become a common design methodology. Altera provides parameterizable megafunctions that are optimized for Altera® device architectures. Using megafunctions instead of coding your own logic saves valuable design time. Additionally, the Altera-provided functions offer more efficient logic synthesis and device implementation. You can scale the size of the megafunction by setting various parameters.

The advantage of floating-point numbers is that they can represent a much larger range of values. In a fixed-point number representation, the radix point is always at the same location. While the convention simplifies numeric operations and conserves memory, it places a limit on the magnitude and precision of the number representation. In situations that require a large range of numbers or high resolution, a relocatable radix point is desirable. Very large and very small numbers can be represented in a floating-point format.

Features

The Floating Point Exponent (ALTFP_EXP) megafunction calculates the exponential value of a given input. In addition, it offers the following features:

- Support for floating-point formats in single, double, and single-extended precision
- Input support for not-a-number (NaN), infinity, zero, and normal numbers
- Optional exception handling output ports such as `nan`, `overflow`, `underflow` and `zero`
- Optional input ports including asynchronous clear (`aclr`) and clock enable (`clk_en`)
- Support for round-to-nearest-even rounding mode

General Description

The ALTFP_EXP megafunction follows the IEEE-754 standard for floating-point exponential and defines the following:

- The formats for representing floating-point numbers
- The representations of special values (zero, infinity, denormal numbers, and bit combinations that do not represent a number [NaN])

The IEEE-754 standard also defines four formats for floating-point numbers. The four formats are: single precision, double precision, single-extended precision, and double-extended precision. The most commonly used floating-point formats are single precision and double precision. The ALTFP_EXP megafunction only supports three formats: single precision, double precision, and single-extended precision.

All of the floating-point formats have binary patterns, as shown in Figure 1–1. In this figure:

- *S* represents a sign bit
- *E* represents an exponent field
- *M* is for the mantissa (part of a logarithm, or fraction) field

For a normal floating-point number, the leading 1 is always implied (for example, binary 1.0011 of decimal 1.1875 is stored as 0011 in the mantissa field). This format can save the mantissa field from using an extra bit to represent the leading 1. However, the leading bit for a denormal number can be either 0 or 1. For zero, infinity, and NaN, the mantissa field does not have a leading 1 implied or any explicit leading bit.

Figure 1–1. IEEE-754 Floating-Point Format

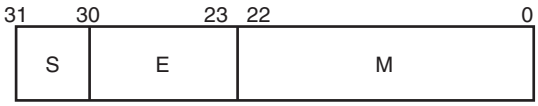
Floating-Point Format

This section describes the formats for single precision, double precision, and single-extended precision.

Single Precision

In single precision, the most significant bit is a sign bit, followed by 8 intermediate bits to represent an exponent, and 23 least significant bits (LSBs) to represent the mantissa. As a result, the total width for single precision is 32 bits. The bias for single precision is 127. Refer to Figure 1–2.

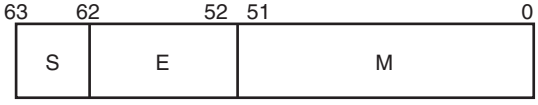
Figure 1–2. Single Precision Representation



Double Precision

In double precision, the most significant bit is a sign bit, followed by 11 intermediate bits to represent an exponent, and 52 LSBs to represent the mantissa. As a result, the total width for double precision is 64 bits. The bias for double precision is 1023. Refer to Figure 1–3.

Figure 1–3. Double Precision Representation



Single-Extended Precision

In single-extended precision, the most significant bit (MSB) is a sign bit. However, the exponent and mantissa fields do not have fixed widths. The width of the exponent field must be a minimum of 11 bits and less than the width of the mantissa field. The width of the mantissa field must be a minimum of 31 bits. The sum of widths for the sign bit, exponent field, and mantissa field must be a minimum of 43 bits and a maximum of 64 bits. The bias for single-extended precision is unspecified in the IEEE-754 standard. In this megafunction, a bias of $2^{(WIDTH_EXP-1)}-1$ is assumed for single-extended precision.

Special Case Numbers

Table 1–1 shows the special case numbers defined by the IEEE-754 standard and the data bit representations.

Table 1–1. Special Case Numbers in IEEE-754 representation

Meaning	Sign Field	Exponent Field	Mantissa Field
Zero	Don't care	All 0's	All 0's
Positive Denormalized	0	All 0's	Non-zero
Negative Denormalized	1	All 0's	Non-zero
Positive Infinity	0	All 1's	All 0's
Negative Infinity	1	All 1's	All 0's
Not-a-Number (NaN)	Don't care	All 1's	Non-zero

Rounding

In the IEEE-754 standard, there are four types of rounding modes: round-to-nearest-even, round-toward-zero, round-toward-positive-infinity, and round-toward-negative-infinity. The most commonly used rounding mode is round-to-nearest-even. However, this megafunction supports only round-to-nearest-even mode. With round-to-nearest-even, the result is rounded to the nearest floating-point number. If the result is exactly halfway between two floating-point numbers, it is rounded so that the LSB becomes a zero, which is even.

Exception Handling

Four optional exception signals are provided: `nan`, `overflow`, `underflow`, and `zero`. For more information about exception ports, refer to Chapter 3, Specifications.

Truth Table

Table 1–2 shows the truth table for the exponential operation.

Table 1–2. Truth Table for the Exponential Operations

DATAA[]	Calculation	RESULT[]	NaN	Overflow	Underflow	Zero
Normal	e^{data}	Normal	0	0	0	0
Normal	e^{data}	Infinity	0	1	0	0
Normal (numbers of small magnitudes)	e^{data}	1	0	0	1	0
Normal (negative numbers of large magnitudes)	e^{data}	0	0	0	1	0
Denormal *(1)*	e^{0}	1	0	0	0	0
Zero	e^{0}	1	0	0	0	0
Infinity (+)	$e^{+\infty}$	Infinity	0	0	0	0
Infinity (-)	$e^{+\infty}$	0	0	0	0	1
NaN	-	NaN	1	0	0	0

Note to Table 1–2:

(1) Any denormal input is treated as a zero before going through the Exponential process.

Resource Utilization and Performance

Table 1–3 shows the resource utilization and performance information for the ALTFP_EXP megafunction.

Table 1–3. ALTFP_EXP Resource Usage and Performance for Stratix III Devices *(Note 1)*

Precision	Output Latency	Logic usage				f_{MAX} (MHz)
		ALUTs	Registers	18-Bit DSP	Memory	
Single	17	527	900	19	0	274.07
Double	25	2,905	2,285	58	0	205.32

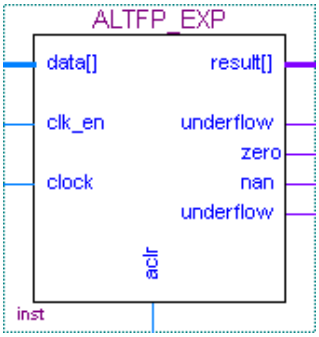
Note to Table 1–3:

(1) You can get the usage and performance information by compiling your design using the Quartus II software. The information in this table is valid and accurate in the Quartus II® software version 8.1.

MegaWizard Plug-In Manager Page Descriptions

Figure 2–1 shows the ports for the ALTFP_EXP megafunction.

Figure 2–1. ALTFP_EXP Ports Shown in the MegaWizard Plug-In Manager



Start the MegaWizard® Plug-In Manager in one of the following ways:

- On the Tools menu, click **MegaWizard Plug-In Manager**.
- When working in the Block Editor, from the Edit menu, click **Insert Symbol as Block** or right-click in the Block Editor, point to **Insert**, and click **Symbol as Block**. In the Symbol window, click **MegaWizard Plug-In Manager**.
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt:

  `qmegawiz↵`

Table 2–1 provides descriptions of the options available on the individual pages of the ALTFP_EXP MegaWizard Plug-In Manager.

Table 2–1. ALTFP_EXP MegaWizard Plug-In Manager Page Options and Description (1 of 3)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
1	Which action do you want to perform?	You can select from the following options: **Create a new custom megafunction variation**, **Edit an existing custom megafunction variation**, or **Copy an existing custom megafunction variation**.

Table 2–1. ALTFP_EXP MegaWizard Plug-In Manager Page Options and Description (2 of 3)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
2a	Select a megafunction from the list below	Select **ALTFP_EXP** from the **Arithmetic** category.
	Which device family will you be using?	Specify the device family that you want to use.
	Which type of output file do you want to create?	You can choose AHDL(**.tdf**), VHDL(**.vhd**), or Verilog HDL (**.v**) as the output file type.
	What name do you want for the output file?	Specify the name of the output file.
	Return to this page for another create operation	Turn on this option if you want to return to this page to create multiple megafunctions.
3	Currently selected device family	Specifies the device family you chose on page 2a.
	Match project/default	Turn on this option to ensure that the device selected matches the device family that is chosen in the previous page.
	What is the floating-point format?	Select **Single precision** for 32 bits, **Double precision** for 64 bits, and **Single-extended precision** for 43 to 64 bits.
	What is the width of the 'data' input and 'result' output buses?	Specify the width of the buses. The maximum width is 32 bits for single precision and 64 bits for double precision and single-extended precision.
	What is the width of the exponent port?	Specify the width of the exponent field. The maximum width is 30 bits.
	Mantissa width = (data width) - (exponent width) - 1	This option is not a user-specified option. The value is calculated automatically when the widths of the exponent field and input buses are specified.
	Output latency in clock cycles	Specify the latency for the result output in clock cycles. The value is fixed at 17 clock cycles for single precision and 25 clock cycles for double precision. For single-extended precision, the latency is 22 clock cycles for mantissa widths of 31 to 38 and 25 for mantissa widths of 29 to 51.
4	Optional Inputs	Two optional input ports are available: asynchronous clear port (`aclr`) and clock enable port (`clk_en`).
	Optional Outputs	Four optional exception or output ports are available: `nan`, `overflow`, `underflow`, and `zero`.

© December 2008 Altera Corporation

Table 2–1. ALTFP_EXP MegaWizard Plug-In Manager Page Options and Description (3 of 3)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
5	Generate netlist	Turn on this option if you want to generate a netlist for your third-party EDA synthesis tool to estimate the timing and resource usage of the megafunction. If you turn on this option, a netlist file (**_syn.v**) is generated. This file is a representation of the customized logic used in the Quartus® II software and provides the connectivity of the architectural elements in the megafunction but may not represent true functionality.
6	Summary Page	Specify the types of files to be generated. The Variation file (<*function name*>**.v**) contains wrapper code in the language you specified on page 2a and is automatically generated. The Quartus II IP file (<*function name*>**.qip**) contains links to all files required to compile an IP variation and is automatically generated when you add an IP core to a project. Choose from the following types of files: ■ AHDL Include file (<*function name*>**.inc**) ■ VHDL component declaration file (<*function name*>.**cmp**) ■ Quartus II symbol file (<*function name*>**.bsf)** ■ Instantiation template file (<*function name*>**_inst.v**) ■ Verilog HDL black-box file (<*function name*>**_bb.v)** For more information about the wizard-generated files, refer to Quartus II Help or to the *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*.

Design Example: Exponential of Single Precision Numbers

This design example uses the ALTFP_EXP megafunction to compute the exponential value of single precision numbers. This example uses the MegaWizard Plug-In Manager in the Quartus II software.

Design Files

The design files are available on the Literature page of the Altera website (www.altera.com). The files are located under the following sections:

- On the Quartus II Literature page, expand the **Using Megafunctions** section and then expand the **Arithmetic** section
- User Guides section

Example

Perform the following steps to compute the exponential value of single precision numbers:

1. Open **altfp_exp_DesignExample.zip** and extract **fp_exp_ex.qar**.
2. In the Quartus II software, open **fp_exp_ex.qar** and restore the archive file into your working directory.
3. On the Tools menu, click **MegaWizard Plug-In Manager**. Page 1 of the MegaWizard Plug-In Manager appears.
4. Select **Create a new custom megafunction variation**.
5. Click **Next**. Page 2a of the MegaWizard Plug-In Manager appears.
6. In the MegaWizard Plug-In Manager pages, select or verify the configuration settings shown in Table 2–2. Click **Next** to advance from one page to the next.

Table 2–2. Configuration Settings for ALTFP_EXP Design Example (1 of 2)

MegaWizard Plug-in Manager Page	Configuration Setting	Value
2a	Select a megafunction	ALTFP_EXP
	Which device family will you be using?	Stratix III
	Which type of output file do you want to create	Verilog HDL
	What name do you want for the output file?	fp_exp_ex
3	Currently selected device family	Stratix III
	Match project/default	Selected
	What is the floating-point format?	Single precision (32 bits)
	What is the width of the ‘data’ input and ‘result’ output buses?	32 bits
	What is the width of the exponent port?	8 bits
	Mantissa width = (data width) - (exponent width) - 1	23 bits
	Output latency in clock cycles	17

Table 2–2. Configuration Settings for ALTFP_EXP Design Example (2 of 2)

MegaWizard Plug-in Manager Page	Configuration Setting	Value
4	Optional inputs	Select all
	Optional outputs	Select all
5	Generate netlist	Not selected
6	Variation file	Selected
	Quartus II IP file	Selected
	Quartus II symbol file	Not selected
	Instantiation template file	Not selected
	Verilog HDL black-box file	Selected
	AHDL Include file	Not selected
	VHDL component declaration file	Not selected

7. Click **Finish**.

If the **Quartus II IP Files** dialog box appears with an option to add the Quartus II IP file (.**qip**) to your project, click **Yes**.

The ALTFP_EXP module is now built.

Functional Simulation in the ModelSim-Altera Software

Simulate the Floating-Point Exponent design in the ModelSim®-Altera software to generate a waveform display of the device behavior.

You need to be familiar with the ModelSim-Altera software before trying out the design example. If you are unfamiliar with the ModelSim-Altera software, refer to the ModelSim support page on the Altera website (www.altera.com). On this support page, there are links to such topics as installation, usage, and troubleshooting.

Set up and simulate the design in the ModelSim-Altera software by performing the following steps:

1. Unzip the **altfp_exp_ex_msim.zip** file to any working directory on your PC.
2. Start the ModelSim-Altera software.
3. On the File menu, click **Change Directory**.
4. Select the folder in which you unzipped the files.
5. Click **OK**.
6. On the Tools menu, click **Execute Macro**.
7. Select the **fp_exp_ex.do** file and click **Open**. The **fp_exp_ex.do** file is a script file for the ModelSim-Altera software to automate all the necessary settings for the simulation.
8. Verify the results shown in the Waveform Viewer window.

You can rearrange signals, remove signals, add signals, and change the radix by modifying the script in **fp_exp_ex.do** accordingly.

Figure 2–2 and Figure 2–3 show the expected simulation results in the ModelSim-Altera software.

Figure 2–2. ModelSim Simulation Waveforms (Input Data)

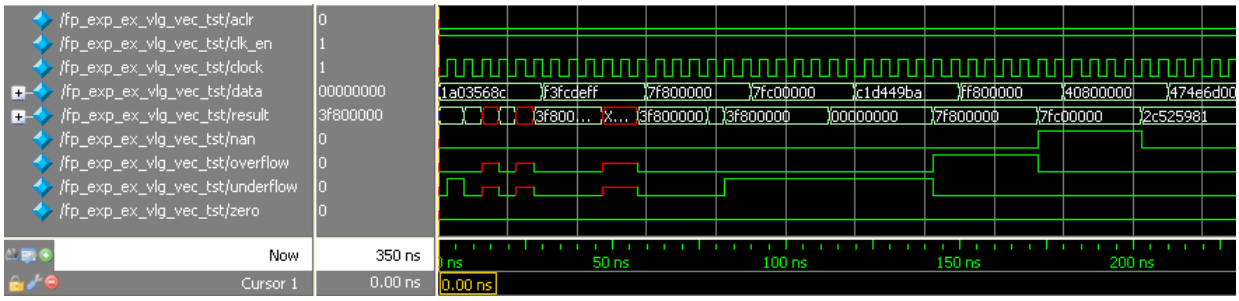
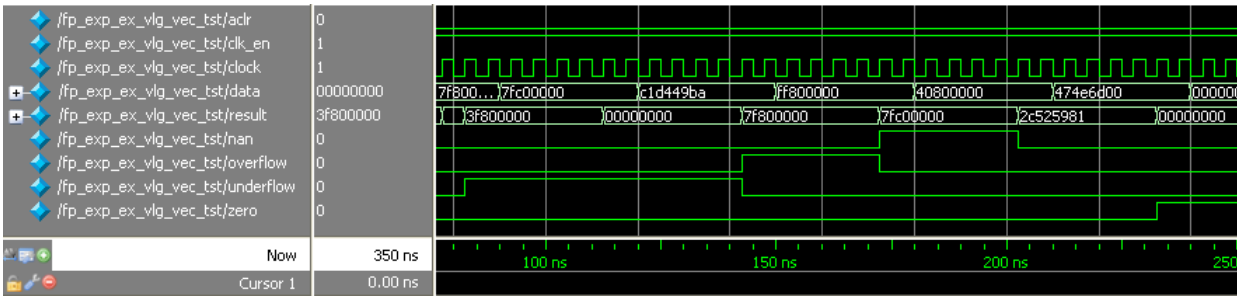


Figure 2–3. ModelSim Simulation Waveforms (Output Data)



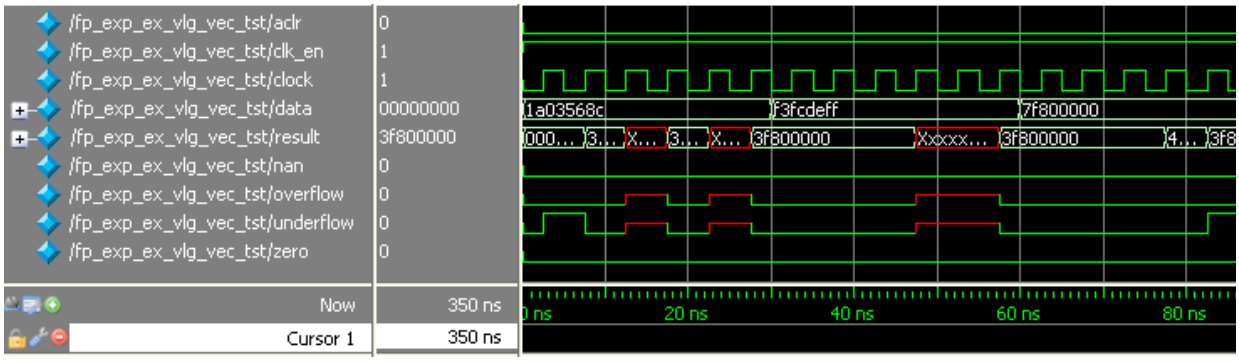
Understanding the Simulation Results

The design example implements a floating-point exponential for single precision numbers. The optional input ports (`clk_en` and `aclr`) and all four exception handling output ports (`nan`, `overflow`, `underflow`, and `zero`) are enabled.

The latency is fixed at 17 clock cycles; therefore, every exponential operation outputs the results 17 clock cycles later.

An undefined value is present on the `result[]`, `overflow`, and `underflow` ports at start-up. This undefined value is generated due to the behavior of the system during start-up. This value is insignificant and can be ignored. Refer to Figure 2–4.

Figure 2–4. Floating-Point Exponential at Power-Up



The result of the first input value is seen on the `result[]` port 17 clock cycles after start-up. The input value of 1A03568Ch is a very small number; therefore, it can be seen as a value that is approaching zero. At 82.5 ns, the result approaches 1 (which is represented by 3F800000). Exponential operations carried out on numbers of very small magnitudes result in a 1 and assert the underflow flag. Refer to Figure 2–5.

Figure 2–5. Exponential of a Normal Value (of a Very Small Magnitude) Results in 1

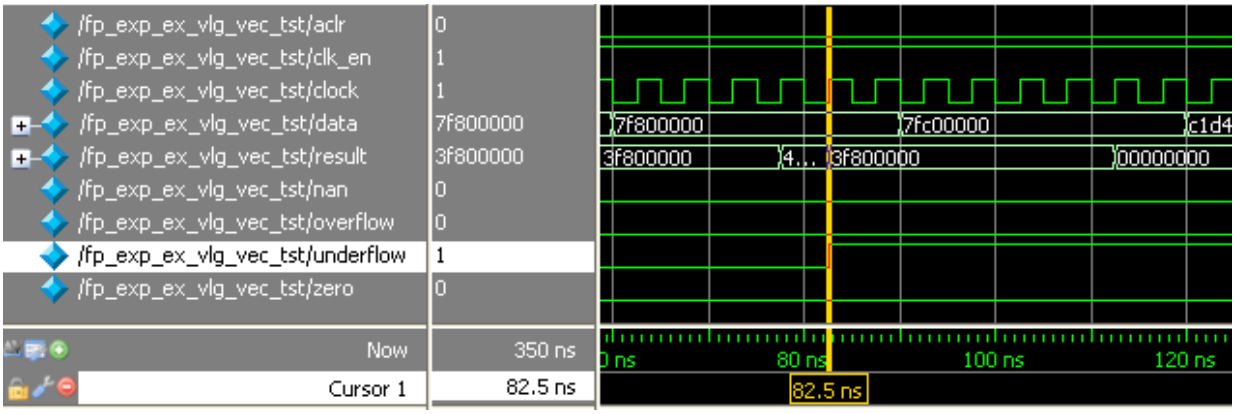
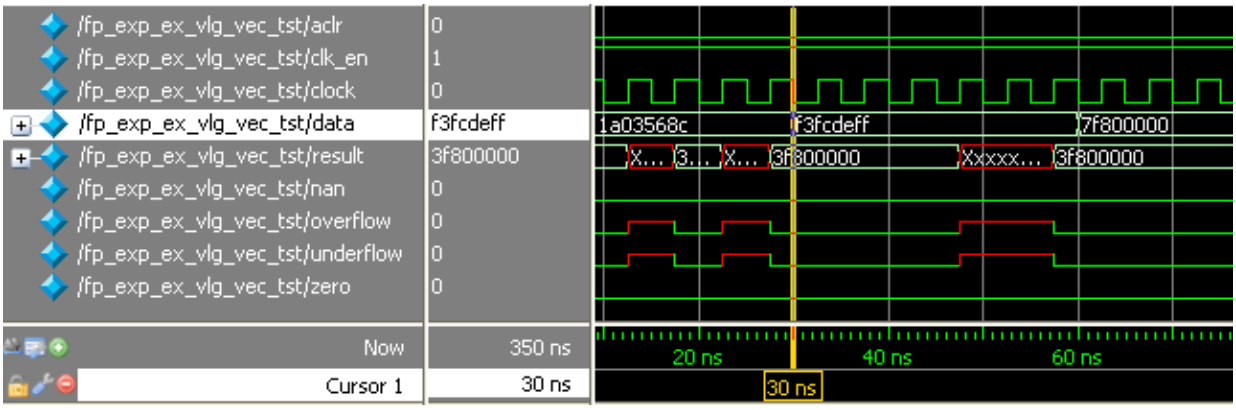


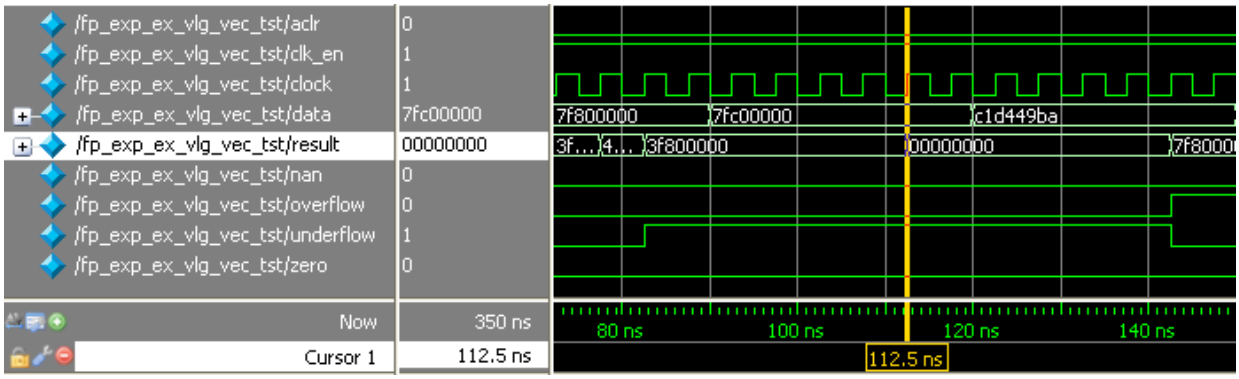
Figure 2–6 shows a normal negative value of a very large magnitude on the `data[]` port at 30 ns.

Figure 2–6. Floating-Point Input at 30 ns



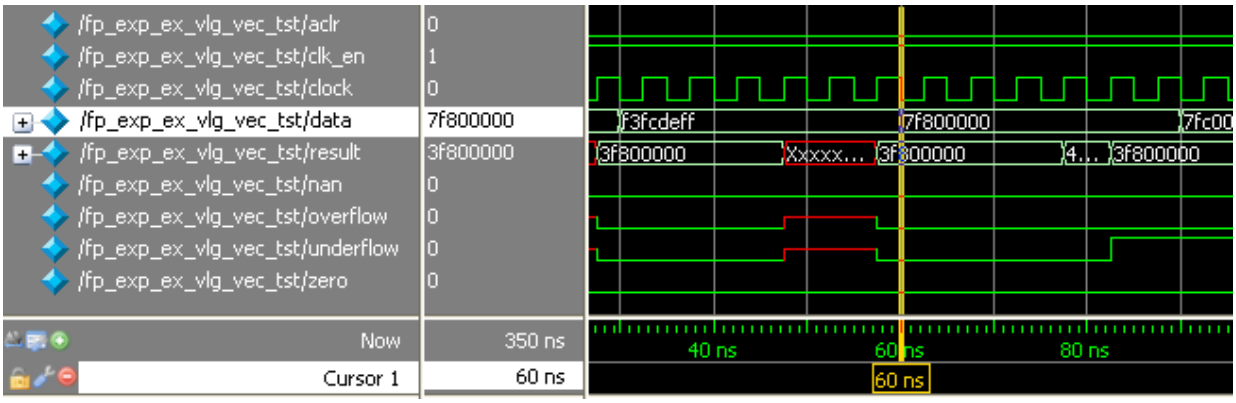
The outcome of the exponential operations on negative numbers of very large magnitudes approaches zero, is seen on the `result[]` port at 112.5 ns. The `underflow` port remains asserted. Refer to Figure 2–7.

Figure 2–7. Exponential of a Normal Negative Value (of a Very Large Magnitude) Results in 0



At 60 ns, a positive infinite value is seen on the `data[]` port. Refer to Figure 2–8.

Figure 2–8. Floating-Point Input at 60 ns



The operation on positive infinite values results in infinity, as can be seen at 142.5 ns. The `underflow` flag deasserts, while the `overflow` flag becomes high. Refer to Figure 2–9.

Figure 2–9. Exponential of an Infinite Value Results in Infinity

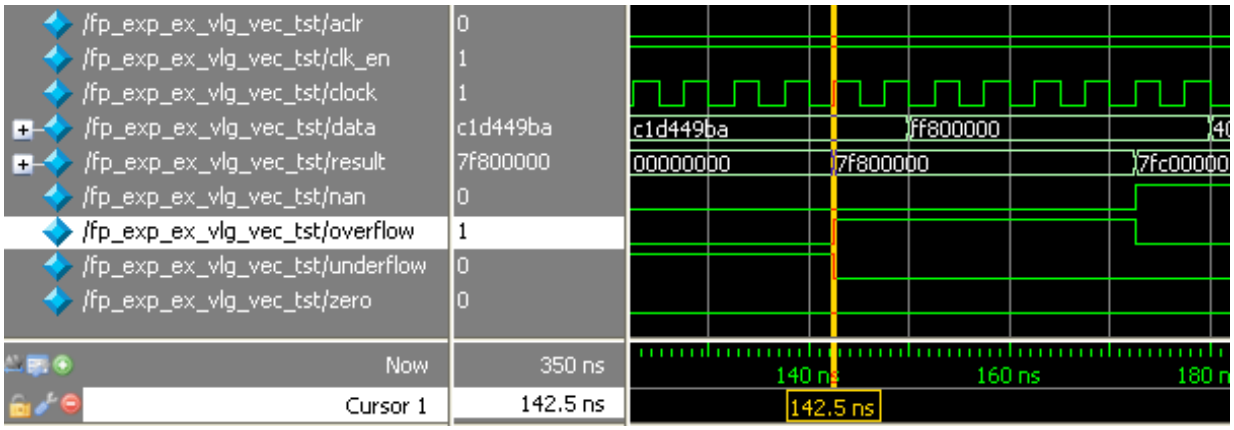
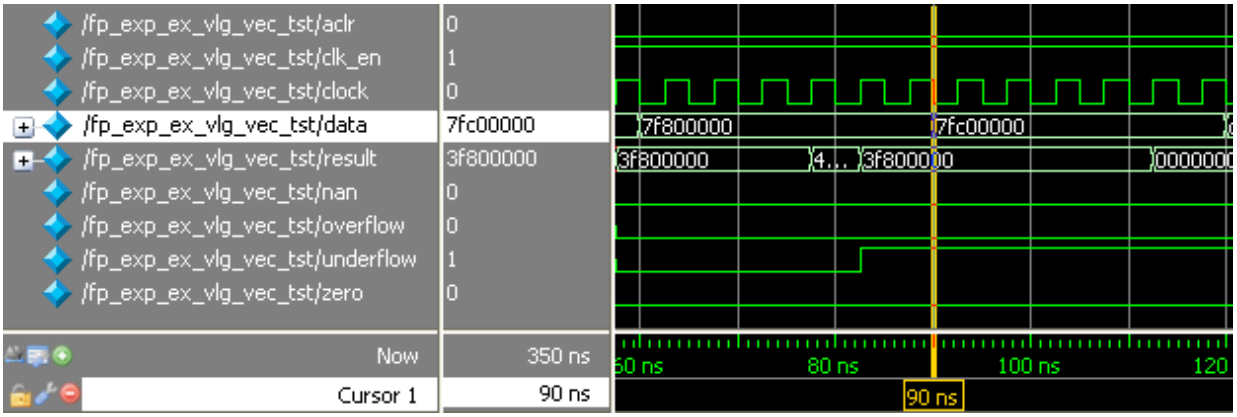


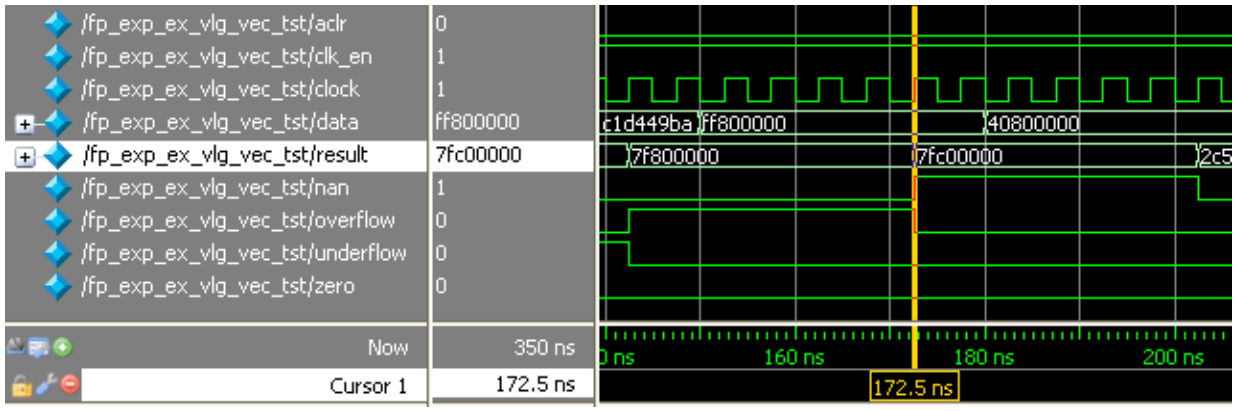
Figure 2–10 shows a not-a-number (NaN) on the `data[]` port at 90 ns.

Figure 2–10. Floating-Point Input at 90 ns



The outcome of the exponential is sent to the `result[]` port at 172.5 ns. The exponential of a NaN results in a NaN, so the `nan` port is asserted. Refer to Figure 2–11.

Figure 2–11. Exponential of a NaN Results in a NaN



At 120 ns, the value on the `data[]` port is a normal value. Refer to Figure 2–12.

Figure 2–12. Floating-Point Input at 120 ns

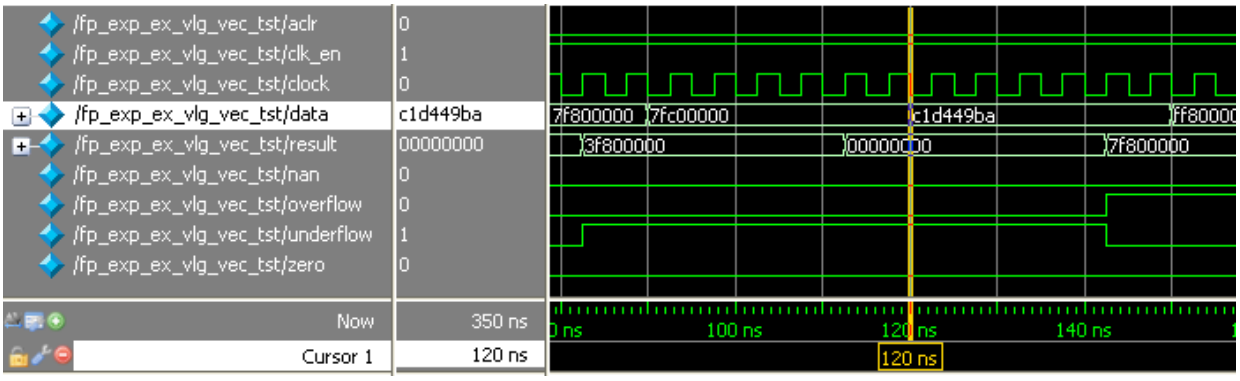
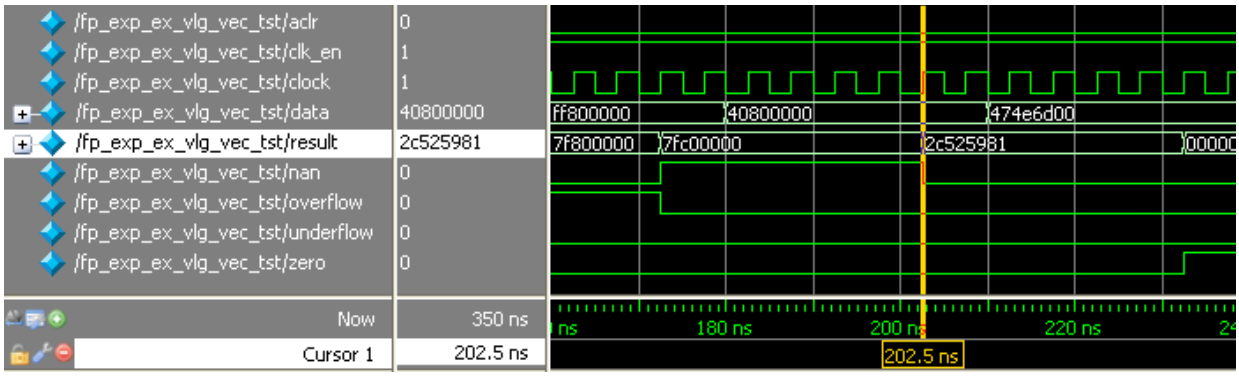


Figure 2–13 show that at 202.5 ns, the exponential of a normal value results in a normal value and causes the `nan` port that was asserted during the previous operation to deassert.

Figure 2–13. Exponential of a Normal Value Results in a Normal Value



For more information about the floating-point exponential operation and the behavior of the output ports, refer to Table 1–2 on page 1–4.

Ports and Parameters

The parameter details are only relevant for users who bypass the MegaWizard® Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in their design. The details of these parameters are hidden from MegaWizard Plug-In Manager interface users.

Refer to the latest version of the Quartus® II Help for the most current information about the ports and parameters of this megafunction.

CAUTION

Altera strongly recommends that you use the MegaWizard Plug-In Manager for complex megafunctions. The MegaWizard Plug-In Manager ensures that you set all megafunction parameters properly.

Table 3–1 shows the input ports, Table 3–2 shows the output ports, and Table 3–3 shows the ALTFP_EXP megafunction parameters.

Input Ports

Table 3–1. ALTFP_EXP Megafunction Input Ports

Port Name	Required	Description	Comments
`aclr`	No	Asynchronous clear	When the `aclr` port is asserted high, the function is asynchronously cleared.
`clk_en`	No	Clock enable	When the `clk_en` port is asserted high, an exponential value operation takes place. When the signal is low, no operation occurs and the outputs remain unchanged.
`clock`	Yes	Clock input	The `clock` port is the clock input to the ALTFP_EXP megafunction.
`data[]`	Yes	Data input	Floating-point input data. The most significant bit (MSB) is the sign, the next most significant bits are the exponent, and the mantissa occupies the least significant bits (LSBs). This input port size is the total width of sign bit, exponent bits, and mantissa bits.

Output Ports

Table 3–2. ALTFP_EXP Megafunction Output Ports (1 of 2)

Port Name	Required	Description	Comments
`result[]`	Yes	The floating point exponential result of the input on `data[]`.	The floating-point result. The MSB is the sign, the next most significant bits are the exponent, and the mantissa occupies the LSBs. The size of this port is the total width of sign bit, exponent bits, and mantissa bits.
`overflow`	No	Optional overflow exception output.	Asserted when the result of the operation (after rounding) is infinite.

Table 3–2. ALTFP_EXP Megafunction Output Ports (2 of 2)

Port Name	Required	Description	Comments
underflow	No	Optional underflow exception output.	Asserted when the result of the exponential approaches 1 (from numbers of a very large magnitude) or when the result approaches 0 (from negative numbers of a very large magnitude).
zero	No	Optional zero exception output.	Asserted when the value at the result[] port is zero.
nan	No	Optional Not-a-Number (NaN) exception output.	Asserted when an invalid operation occurs. Any operation involving a NaN value also asserts the nan port.

Parameters

Table 3–3. ALTFP_EXP Megafunction Parameters

Parameter Name	Type	Required	Comments
WIDTH_EXP	Integer	Yes	Specifies the precision of the exponent. If omitted, the default value for WIDTH_EXP is 8. The bias of the exponent is always set to $2^{(WIDTH_EXP\ -1)}$ -1 (for example, 127 for single precision floating-point format and 1023 for double precision floating-point format). The WIDTH_EXP value must be 8 for single precision floating-point format and 11 for double precision, and a minimum of 11 for single-extended precision. The WIDTH_EXP value must be less than the WIDTH_MAN value. The sum of WIDTH_EXP and WIDTH_MAN must be less than 64.
WIDTH_MAN	Integer	Yes	Specifies the precision of the mantissa. If omitted, the default value for WIDTH_MAN is 23. When the WIDTH_EXP value is 8 and the floating-point format is single precision, the WIDTH_MAN value must be 23. Otherwise, the value of WIDTH_MAN must be a minimum of 31. The WIDTH_MAN value must be greater than the WIDTH_EXP value. The sum of WIDTH_EXP and WIDTH_MAN must be less than 64.
PIPELINE	Integer	Yes	Specifies the amount of latency in clock cycles used in the megafunction. Create the ALTFP_EXP megafunction. Acceptable pipeline values are 17, 22, and 25 cycles of latency. Create the ALTFP_EXP megafunction variation with the MegaWizard Plug-In Manager to calculate the value for this parameter.
DEVICE_FAMILY	String	Yes	This parameter is used for modeling and behavioral simulation purposes. Create the ALTFP_EXP megafunction variation with the MegaWizard Plug-In Manager to determine the value for this parameter.

Document Revision History

The table below displays the revision history for the chapters in this User Guide.

Date	Document Version	Changes Made
December 2008	1.0	Initial release

Referenced Documents

This user guide references the following documents:

- *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook*
- *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*
- *Simulation* section in volume 3 of the *Quartus II Handbook*
- *Synthesis* section in volume 1 of the *Quartus II Handbook*

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, file names, file name extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design*.
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.